

```
In [23]: import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
pd.options.plotting.backend = "plotly"
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
```

```
In [93]: threshold=pd.Timedelta(2,unit="m")
```

```
In [53]: #threshold
monday=pd.read_csv("Monday (1).csv",index_col=[0])
monday["timestamp"]=pd.to_datetime(monday["index"])
monday["count"]=[1 for i in range(0,len(monday))]
monday.info()
monday_timestamp=[]

count_bunches=0
monday_sorted=monday.sort_values(by="timestamp",ascending=True)
monday_sorted.reset_index(inplace=True)
del monday_sorted["level_0"]

for i in range(0,len(monday)):
    for j in range(i+1,len(monday)):
        if(monday_sorted["timestamp"][j]-monday_sorted["timestamp"][i] <= threshold)
            count_bunches+=1
            monday_timestamp.append(monday_sorted["timestamp"][j])

        continue
    else:
        break
print("No of time bunching happended : ",count_bunches)
print("probability of bunching taking place: ",count_bunches/len(monday))
print("With in time interval of: ",min(monday["timestamp"]), " to ",max(monday["times
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 40 entries, 0 to 39
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   index       40 non-null    object
1   timestamp   40 non-null    datetime64[ns]
2   count       40 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 1.2+ KB
No of time bunching happended : 10
probability of bunching taking place: 0.25
With in time interval of: 2022-03-07 14:54:31.462434 to 2022-03-07 20:24:22.695238
```

```
In [54]: #threshold
tuesday=pd.read_csv("tuesday.csv",index_col=[0])
tuesday["timestamp"]=pd.to_datetime(tuesday["index"])
tuesday["count"]=[1 for i in range(0,len(tuesday))]
tuesday.info()
tuesday_timestamp=[]
```

```

count_bunches=0
tuesday_sorted=tuesday.sort_values(by="timestamp",ascending=True)
tuesday_sorted.reset_index(inplace=True)
del tuesday_sorted["level_0"]

for i in range(0,len(tuesday)):
    for j in range(i+1,len(tuesday)):
        if(tuesday_sorted["timestamp"][j]-tuesday_sorted["timestamp"][i] <= threshold):
            count_bunches+=1
            tuesday_timestamp.append(tuesday_sorted["timestamp"][j])

        continue
    else:
        break
print("No of time bunching happended : ",count_bunches)
print("probability of bunching taking place: ",count_bunches/len(tuesday))
print("With in time interval of: ",min(tuesday["timestamp"]), " to ",max(tuesday["tim

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 36 entries, 0 to 35
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   index       36 non-null    object
1   timestamp   36 non-null    datetime64[ns]
2   count       36 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 1.1+ KB
No of time bunching happended : 6
probability of bunching taking place: 0.16666666666666666
With in time interval of: 2022-03-15 14:32:09.633333 to 2022-03-15 19:06:07.89597
7

```

In [55]:

```

#threshold
wednesday=pd.read_csv("Wednesday.csv",index_col=[0])
wednesday["timestamp"]=pd.to_datetime(wednesday["index"])
wednesday["count"]=[1 for i in range(0,len(wednesday))]
wednesday.info()
wednesday_timestamp=[]

count_bunches=0
wednesday_sorted=wednesday.sort_values(by="timestamp",ascending=True)
wednesday_sorted.reset_index(inplace=True)
del wednesday_sorted["level_0"]

for i in range(0,len(wednesday)):
    for j in range(i+1,len(wednesday)):
        if(wednesday_sorted["timestamp"][j]-wednesday_sorted["timestamp"][i] <= threshold):
            count_bunches+=1
            wednesday_timestamp.append(wednesday_sorted["timestamp"][j])

        continue
    else:
        break
print("No of time bunching happended : ",count_bunches)
print("probability of bunching taking place: ",count_bunches/len(wednesday))
print("With in time interval of: ",min(wednesday["timestamp"]), " to ",max(wednesday[

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 45 entries, 0 to 44
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype

```

```

-----
0  index      45 non-null    object
1  timestamp  45 non-null    datetime64[ns]
2  count      45 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 1.4+ KB
No of time bunching happended : 8
probability of bunching taking place: 0.17777777777777778
With in time interval of: 2022-03-16 14:14:48.233058 to 2022-03-16 19:12:16.43333
3

```

In [56]:

```

#threshold
thursday=pd.read_csv("Thursday (1).csv",index_col=[0])
thursday["timestamp"]=pd.to_datetime(thursday["index"])
thursday["count"]=[1 for i in range(0,len(thursday))]
thursday.info()
count_bunches=0
thursday_timestamp=[]
thursday_sorted=thursday.sort_values(by="timestamp",ascending=True)
thursday_sorted.reset_index(inplace=True)
del thursday_sorted["level_0"]

for i in range(0,len(thursday)):
    for j in range(i+1,len(thursday)):
        if(thursday_sorted["timestamp"][j]-thursday_sorted["timestamp"][i] <= thresh
            count_bunches+=1
            thursday_timestamp.append(thursday_sorted["timestamp"][j])
            continue
        else:
            break
print("No of time bunching happended : ",count_bunches)
print("probability of bunching taking place: ",count_bunches/len(thursday))
print("With in time interval of: ",min(thursday["timestamp"]), " to ",max(thursday["t

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 59 entries, 0 to 58
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   index       59 non-null    object
1   timestamp   59 non-null    datetime64[ns]
2   count       59 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 1.8+ KB
No of time bunching happended : 10
probability of bunching taking place: 0.1694915254237288
With in time interval of: 2022-03-10 11:31:05 to 2022-03-10 18:45:13.200000

```

In [57]:

```

#threshold
friday=pd.read_csv("Friday.csv",index_col=[0])
friday["timestamp"]=pd.to_datetime(friday["index"])
friday["count"]=[1 for i in range(0,len(friday))]
friday.info()
friday_timestamp=[]

count_bunches=0
friday_sorted=friday.sort_values(by="timestamp",ascending=True)
friday_sorted.reset_index(inplace=True)
del friday_sorted["level_0"]

for i in range(0,len(friday)):
    for j in range(i+1,len(friday)):
        if(friday_sorted["timestamp"][j]-friday_sorted["timestamp"][i] <= threshold)

```

```

count_bunches+=1
friday_timestamp.append(thursday_sorted["timestamp"][j])

        continue
    else:
        break
print("No of time bunching happended : ",count_bunches)
print("probability of bunching taking place: ",count_bunches/len(friday))
print("With in time interval of: ",min(friday["timestamp"]), " to ",max(friday["times

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 60 entries, 0 to 59
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   index       60 non-null    object
 1   timestamp   60 non-null    datetime64[ns]
 2   count       60 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 1.9+ KB
No of time bunching happended : 3
probability of bunching taking place: 0.05
With in time interval of: 2022-03-18 14:19:17.315909 to 2022-03-18 23:04:28.875000
0

```

## Daywise bunching graphs

```

In [88]: fig=px.histogram(monday_timestamp,nbins=10,histnorm="probability",title="Distributio
        labels={"value":"Blue Bus Arrival Timestamp"},template="simple_white")
fig.show()

```

### Distribution of Blue Buses Wednesday



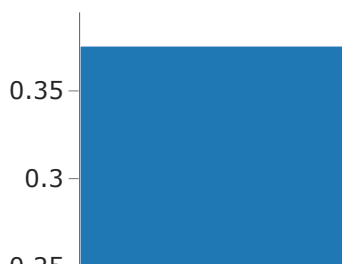
```
In [89]: fig=px.histogram(tuesday_timestamp,nbins=10,histnorm="probability",title="Distributi  
labels={"value":"Blue Bus Arrival Timestamp"},template="simple_white")  
fig.show()
```

Distribution of Blue Buses Wednesday



```
In [90]: fig=px.histogram(wednesday_timestamp,nbins=10,histnorm="probability",title="Distribu  
labels={"value":"Blue Bus Arrival Timestamp"},template="simple_white")  
fig.show()
```

Distribution of Blue Buses Wednesday



```
In [91]: fig=px.histogram(thursday_timestamp,nbins=10,histnorm="probability",title="Distribut  
labels={"value":"Blue Bus Arrival Timestamp"},template="simple_white")  
fig.show()
```

### Distribution of Blue Buses Wednesday



```
In [92]: fig=px.histogram(friday_timestamp,nbins=5,histnorm="probability",title="Distribution  
labels={"value":"Blue Bus Arrival Timestamp"},template="simple_white")  
fig.show()
```

### Distribution of Blue Buses Wednesday





## Whole week bunching statistics

```
In [61]: bunching_timestamp=[]
bunching_timestamp.extend(monday_timestamp)
bunching_timestamp.extend(tuesday_timestamp)
bunching_timestamp.extend(wednesday_timestamp)
bunching_timestamp.extend(thursday_timestamp)
bunching_timestamp.extend(friday_timestamp)
```

```
In [78]: print("No of time bunching happended in the week: ",len(bunching_timestamp))
print("probability of bunching taking place through out the week: ",
      (len(bunching_timestamp)/(len(monday)+len(tuesday)+len(wednesday)+len(thursday)
print("With in time interval of: ",min(bunching_timestamp)," to ",max(bunching_times
```

```
No of time bunching happended in the week: 37
probability of bunching taking place through out the week: 0.15416666666666667
With in time interval of: 2022-03-10 11:52:59.297436 to 2022-03-10 17:37:25.62946
9
```

```
In [84]: fig=px.histogram(bunching_timestamp,nbins=10,histnorm="probability",title="Distribut
labels={"value":"Blue Bus Arrival Timestamp"},template="simple_white")
fig.show()
```

## Distribution of Blue Buses Bunching throughout the Week



n 2



In [1]:

```
!conda install plotly
```

^C

In [ ]: